

## PATENT

REMARKS

This paper is responsive to a final Office action dated April 21, 2005. Claims 1-23 were examined. Claims 12-22 are allowed.

Claims Objections

Claims 9 and 10 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitation so f the base claim and any intervening claims. Applicant appreciates the indication of allowable subject matter, but respectfully submits that the parent claim of claims 9 and 10 is allowable for the reasons provided below.

Art Rejection—35 U.S.C. §102Stone

Claims 1, 4 and 23 are rejected under 35 U.S.C. §102(b) as being anticipated by “A simple and correct shared-queue algorithm using Compare-and-Swap” by Janice M. Stone (hereinafter “Stone”). Applicant respectfully traverses these rejections.

In rejecting claim 1, the Office states that Stone’s algorithm is non-blocking only if a processor fails. However, this statement is incorrect because 1) it is not necessary for a processor to fail for Stone’s algorithm to not be non-blocking, and 2) the Office confuses nondelaying with non-blocking. As stated in Prakash when describing Stone, “a faulty or slow enqueueer can block all the dequeuers.” In addition, Stone states itself several times that it is nondelaying, and not non-blocking. The Office states that Stone is only non-blocking when a processor fails. In addition to this statement being incorrect since a slow enqueueer can also block dequeuers as already stated, the statement by the Office turns the characterization on its head. A failed processor does not characterize an algorithm as non-blocking, but an algorithm, such as Stone’s algorithm, is not non-blocking because it allows a failed or slow enqueueer to block other operations. A non-blocking algorithm does not block, despite a slow or failed enqueueer. Stone states at p. 503 that “starvation is possible: it is possible for a processor to fail all its attempts to modify a shared pointer.” Stone also states at p. 503 that “a particular

## PATENT

processor could lose all its Compare-and-Swap contentions.” Hence, it can be seen that Stone repeatedly identifies its own algorithm as not non-blocking.

In rejecting claim 23, the Office identifies Stone’s CSDBL as the atomic operation that disambiguates a retry state from a boundary condition. The statements by the Office mischaracterize the instance of the CSDBL operation in Stone’s Dequeue procedure as well as incorporating functions of other statements of the Dequeue procedure into the CSDBL operation instances. The Office refers to the CSDBL for the retry state and then refers to a separate conditional state for the boundary condition. A CSDBL in the Dequeue procedure of Stone occurs 3 times. The CSDBL operation is utilized to set the shared tail pointer to null if there is only one item in the queue; set the shared head pointer; and dequeue the item. Not one of these instances of the CSDBL operation disambiguates a retry state and a boundary condition. The Office blurs together the retry performed by the Dequeue procedure of Stone and the boundary condition detected by the conditional statement. The lines of code relied upon by the Office are external and separate from the CSDBL operation instances in the Dequeue procedure of Stone. Thus, none of the CSDBL operations of Stone disambiguate a retry state and a boundary condition state of concurrent shared object as recited in claim 23.

Prakash

Claims 1 and 3-5 are rejected under 35 U.S.C. §102(b) as being anticipated by “*A Non-blocking Algorithm for Shared Queues Using Compare-and-Swap*” by Sundeep Prakash, et al. (hereinafter “Prakash”).

In rejecting claim 1 with Prakash, the Office seems to disregard the portion of the Enqueue procedure depicted in Figure 5 of Prakash identified by Applicant as disclosing cooperation between enqueueers and dequeuers. The Office states the following:

[T]he Cooperate in dequeuing the object has no program code in the figure. Figure 5 has comments, program code, and explanations throughout. The italicize text that Applicant refers to is not program code. Therefore, the algorithm never helps the dequeue operations or the extension of the helping such that the algorithms are not disjoint.

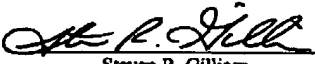
## PATENT

Applicant respectfully refers to page 548 of Prakash which states that "[t]he non-blocking property is achieved by processes cooperating to complete an operation started by one of the processes." Prakash also states at page 549 that Prakash successfully implements the second option, which allows any process to perform subsequent steps of an enqueue or dequeue operation that comprises more than a single step. Hence it can be seen that in addition to the algorithm depicted in Figure 5, Prakash discloses cooperation between enqueueers and dequeuers as the key to Prakash's algorithm allegedly achieving a non-blocking property. Since Prakash implements enqueueers and dequeuers as cooperating, then Prakash cannot disclose or suggest "wherein, at least for those of the valid states other than the one or more boundary condition states, opposing-end ones of the access operations are disjoint" as recited in claim 1.


For at least the reasons above, the independent claims 1 and 23 are allowable over the art of record. In addition, the dependent claims 2 – 11 are at least allowable because they depend from allowable independent claims 1.

Conclusion

In summary, claims 1-23 are in the case. All claims are believed to be allowable over the art of record, and a Notice of Allowance to that effect is respectfully solicited. Nonetheless, if any issues remain that could be more efficiently handled by telephone, the Examiner is requested to call the undersigned at the number listed below.

<b>CERTIFICATE OF MAILING OR TRANSMISSION</b>	
I hereby certify that, on the date shown below, this correspondence is being	
<input type="checkbox"/>	deposited with the US Postal Service with sufficient postage as first class mail and addressed as shown above.
<input checked="" type="checkbox"/>	facsimile transmitted to the US Patent and Trademark Office.
 Steven R. Gilliam	<u>21-Jun-2005</u> Date

Respectfully submitted,

  
Steven R. Gilliam, Reg. No. 51,734  
Attorney for Applicant(s)  
(512) 338-6320  
(512) 338-6301 (fax)

EXPRESS MAIL LABEL: \_\_\_\_\_